# FRAUNHOFER HHI AT TRECVID 2004: SHOT BOUNDARY DETECTION SYSTEM

*Christian Petersohn*

Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut,
Einsteinufer 37, 10587 Berlin, Germany
(petersohn@hhi.fhg.de)

## ABSTRACT

This paper describes the shot boundary detection and determination system developed at the Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut used for the evaluation at TRECVID 2004. The system detects and determines the position of hard cuts, dissolves, fades and wipes. It is very fast and proved to have a very good detection performance.

As input for our system we used luminance pixel values of sub-sampled video data. The hard cut detector uses pixel and edge differences with an adaptive thresholding scheme. Flash detection and slow motion detection lower the false positive rate. Dissolve and Fade detection is done with edge energy statistics, pixel and histogram differences and a linearity measure. Wipe detection works with an evenness factor and double Hough transform. The difference between the submitted runs is basically only different threshold settings in the detectors, resulting in different recall and precision values.

## 1. INTRODUCTION

Huge amounts of video data are produced around the world each day. An automatic video shot detection and determination is important for tasks involving management, analysis and search and retrieval of video data.

This paper gives an overview of the shot detection system developed at the Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut. The different detection steps are described. Results on the TRECVID 2004 test set conclude the paper.

## 2. SYSTEM OVERVIEW

The shot detection system starts with decoding the MPEG-file. We use mpeg2dec (http://libmpeg2.sourceforge.net/) which is the fastest free MPEG-decoder we could find. It is available under GPL. We have done some modifications on the decoder during our research to be able to extract additional information like DC-coefficients and motion vectors. The system used for TRECVID does a full decoding and works afterwards on sub sampled video frames (by factor eight in x and y direction) containing luminance information only. This small input data rate used in the calculation of statistics and in the detectors makes a very fast processing possible.

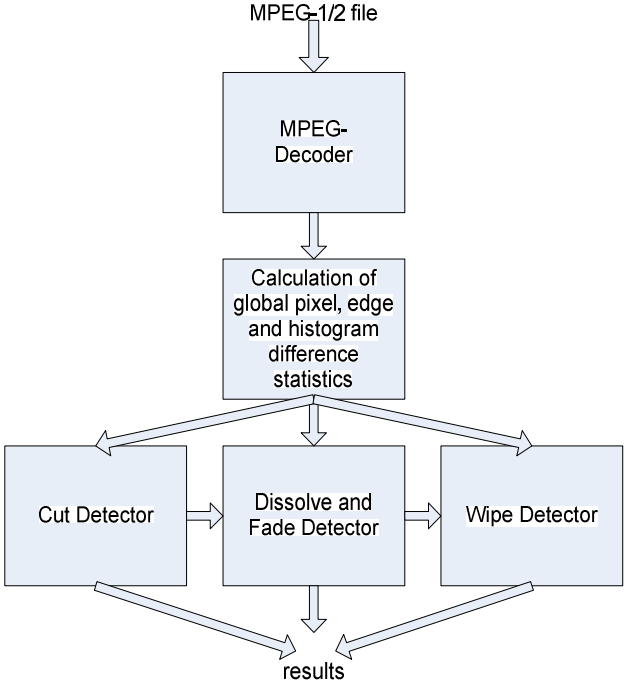



**Fig. 1: System overview**

The calculation of pixel, edge and histogram difference statistics is only done for features one or more detectors need for all frames. Features only needed in special situations (flash detection, e.g.) are calculated when they are actually used. The different detectors themselves are designed to consist of different stages. The first stage always marks shot boundary candidates. The following stages are only used for the candidates of the preceding stage and test whether additional criteria are met. More complex operations are always done in a stage as late as possible.

## 3. HARD CUT DETECTION

For hard cut detection we use edge and pixel differences calculated for consecutive frames after simple motion compensation. To be able to adapt to the varying statistics in a video file adaptive thresholding is used to mark hard cut candidates. Afterwards cut candidates are tested with a flash detector working with pixel and edge differences for frames with different temporal distances around the candidate frame. If their difference is too small the hard cut candidate is rejected. As an additional step we test for slow motion passages before marking candidates as we found that our adaptive thresholding scheme produced too many false positive for sections when only every nth frame changes and the differences between other frames are only caused by noise. Then the difference values are altered for that part before they are used by the adaptive thresholding scheme.

## 4. DISSOLVE AND FADE DETECTION

The dissolve detector consists of six stages [1]:

1. dissolve candidate selection by searching for U-shapes in the edge energy diagram (Fig. 3, left)
2. checking image differences several frames apart and determination of position and duration of the dissolve candidate (Fig. 3, right)
3. checking histogram differences and motion compensated image differences
4. checking dissolve linearity
5. checking dissolve evenness
6. checking global motion with cross correlation

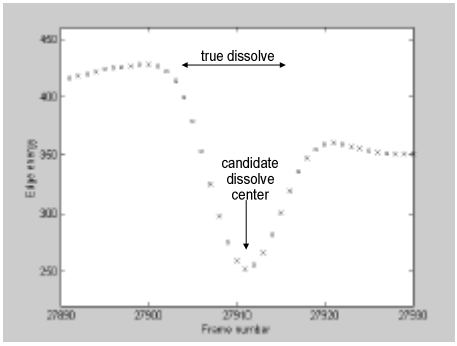



**Fig. 2: edge energy diagram (stage 1)**

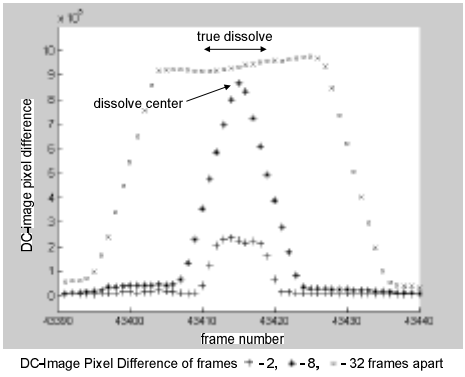



**Fig. 3: image differences curves (stage 2)**

Stage 4, linearity, basically works with three images S, C, E for the dissolve start, center and end and compares pairwise differences using triangle inequality. During a dissolve the difference between S and E should be as large as the sum of the differences between S and C and C and E. Where as during motion etc. the difference between S and E tends to be smaller.

Stage 5 works with an evenness measure we defined which is related to the variance of difference images. Changes in between frames during a dissolve tend to be more evenly distributed than during object motion or object fade out/in (see Fig. 4)

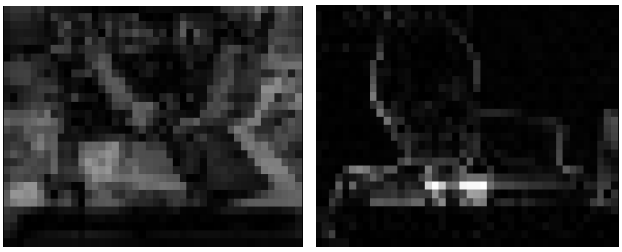


**Fig. 4: Difference image during a dissolve (left) and a false dissolve candidate with object motion and headline fade-out (right)**

During dissolve detection many fades get marked as shot boundary candidates. If edge energy is below a threshold close to zero we mark it as fade.

## 5. WIPE DETECTION

For TRECVID we used an initial version of our wipe detector. For marking wipe candidates it uses an evenness factor to exploit the observation that during a wipe spatial zones of change move thru the image [2].

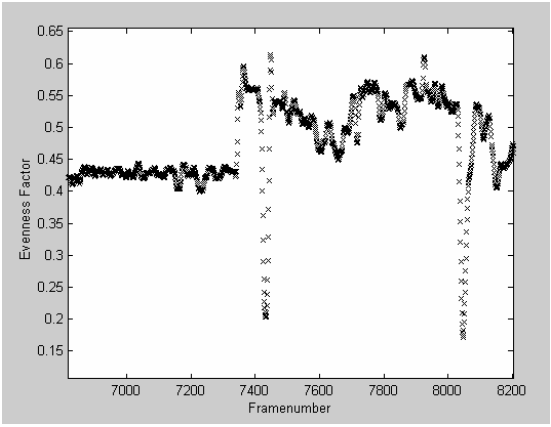



**Fig. 5: Evenness factor curve with two wipes**

In the second stage image differences several frames apart are checked. In the third stage we use a double Hough transform on the set of difference images in the wipe candidate. The first Hough transform detects linear segments in the difference images (boundaries in the images during the wipe). The accumulators in the transformed Hough spaces (Fig. 7) are set to one if a line was found. All Hough spaces for the difference images are added (Fig. 8) and the result is Hough transformed itself. This is done to find patterns in the movement of the boundaries during the wipe.

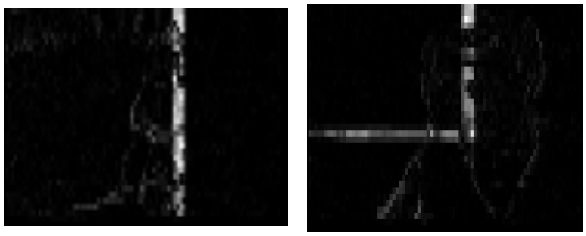


**Fig. 6: difference images during a wipe**

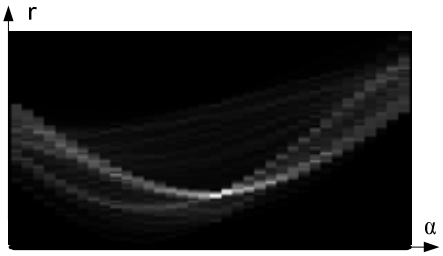
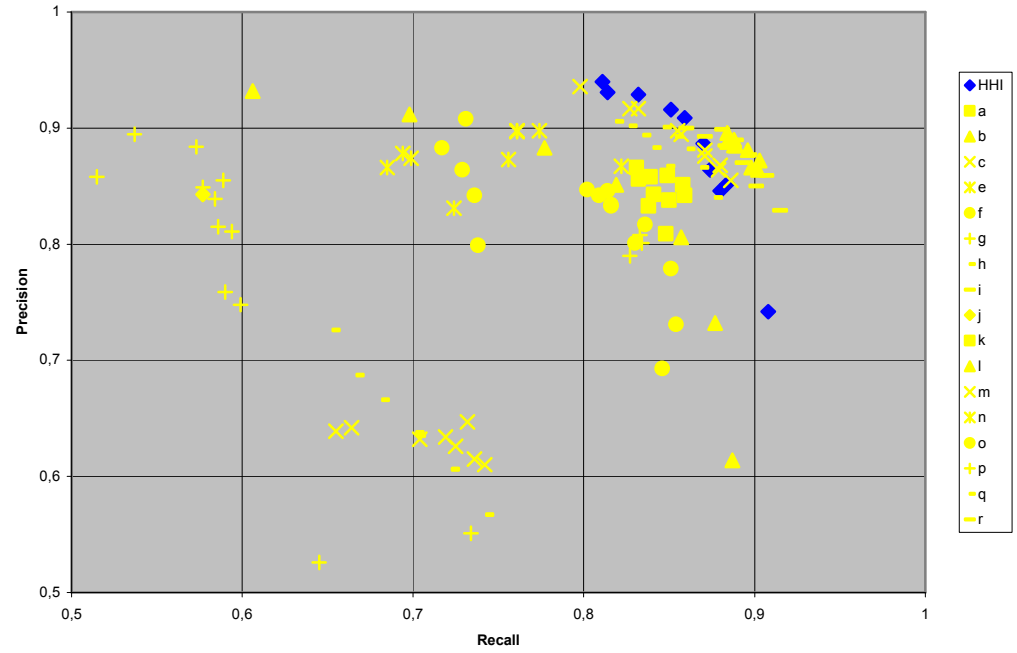



**Fig. 7: Hough transform of difference image**

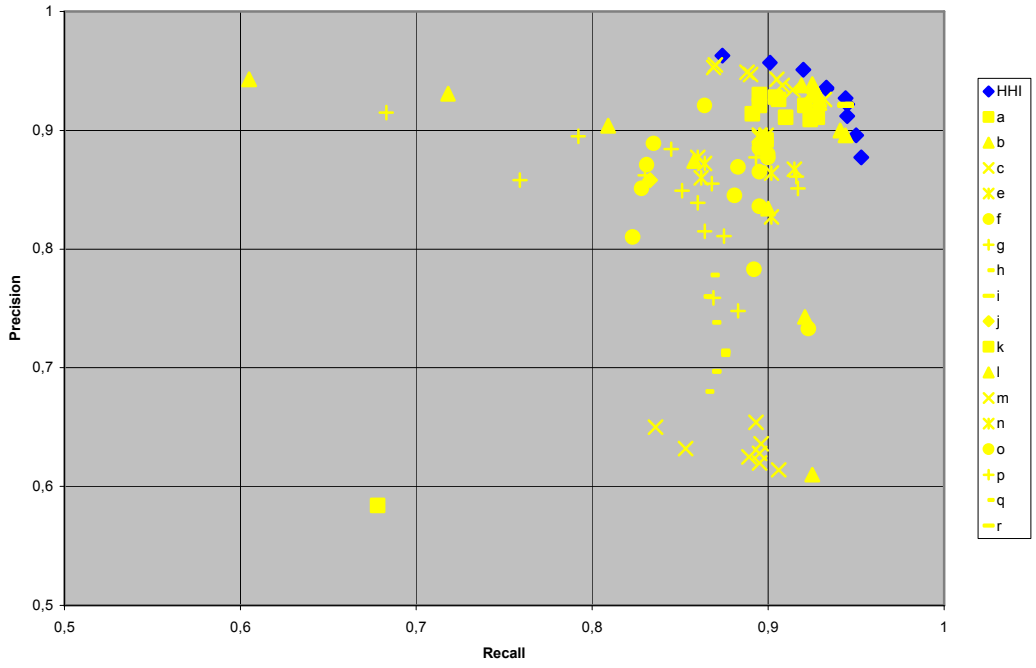

**Fig. 8: Added binarized Hough spaces, input for second Hough transform**

# 7. RESULTS AT TRECVID

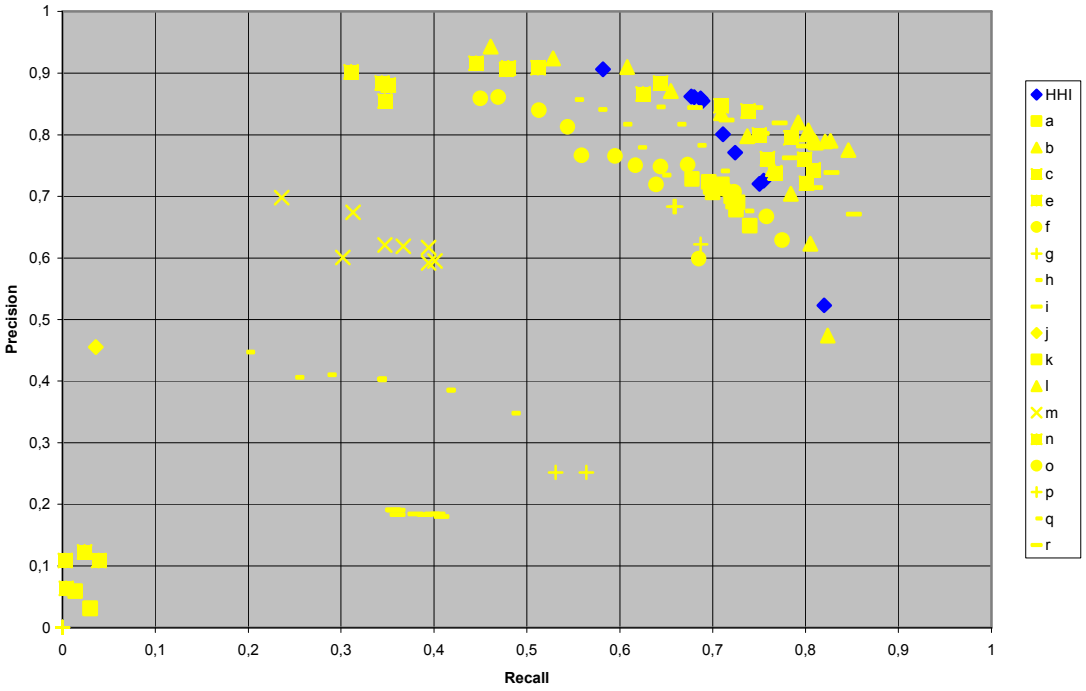
**All Transitions**



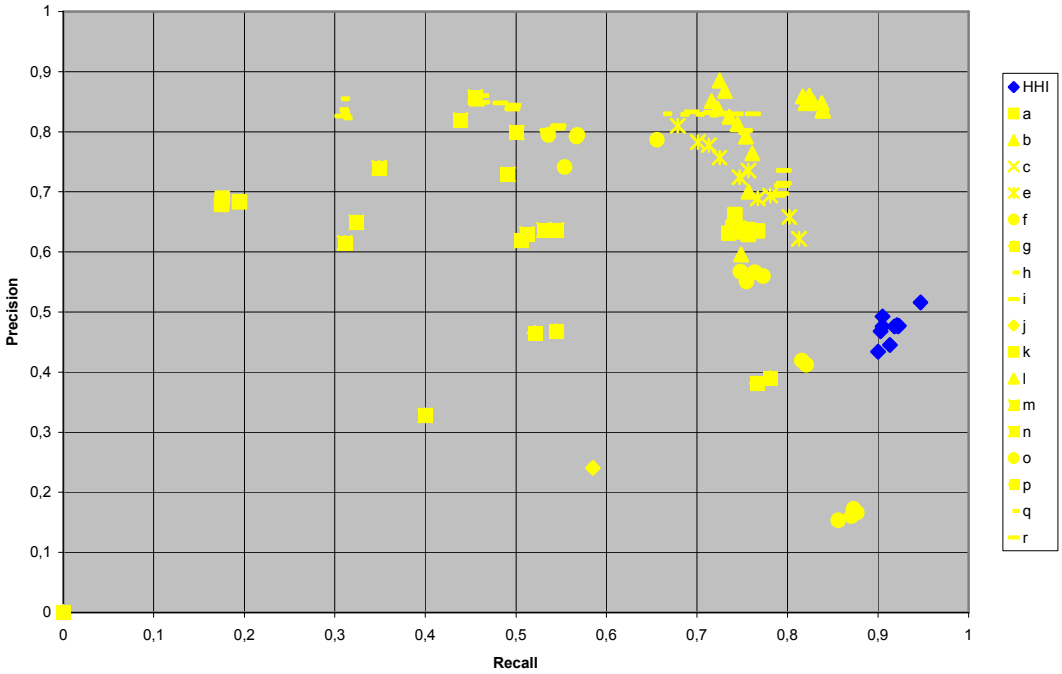

**CUTs**



**Gradual Transitions**




**Frame Recall and Frame Precision for Gradual Transitions**



## COMPUTATIONAL COMPLEXITY

We measured execution times on a PC with P4 Xeon 3,06GHz processor. Decoding times include MPEG-decoding only. Therefore they are independent of parameters used during segmentation and the same for every run. Segmentation times include sub sampling of the frames, extracting and analysing feature statistics.

| | Total time in sec | Decoding time in sec | Segmenta-tion time in sec |
|---|---|---|---|
| Run-1 | 996 | 600 | 396 |
| Run-2 | 993 | 600 | 393 |
| Run-3 | 968 | 600 | 368 |
| Run-4 | 1001 | 600 | 401 |
| Run-5 | 966 | 600 | 366 |
| Run-6 | 984 | 600 | 384 |
| Run-7 | 990 | 600 | 390 |
| Run-8 | 991 | 600 | 391 |
| Run-9 | 1003 | 600 | 403 |
| Run-10 | 1011 | 600 | 411 |

Total real playing time for the shot boundary detection test set is 20614 seconds (618409 frames). That means our system is about 20 times faster than real time on the test set on the computer we used.

## 9. REFERENCES

[1] C. Petersohn. "Dissolve Shot Boundary Determination", To appear in *Proc. IEE European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, London, UK, 2004

[2] C. Petersohn. "Wipe Shot Boundary Determination", To appear in *Proc. SPIE Storage and Retrieval Methods and Applications for Multimedia 2005*, San Jose, CA, 2005